

BAB II

LANDASAN TEORI

Bab ini berisi tentang data dan informasi yang berkaitan dengan pokok permasalahan yang akan diuji, yaitu dengan mendalami dan membaca literatur seputar *Online Public Access Catalog (OPAC)*. Pembahasan juga meliputi referensi lain yang berhubungan dengan ontologi, TF-IDF dan *cosine similarity* yang digunakan sebagai metode dalam menyelesaikan penelitian ini.

2.1 Literatur Review

Penelitian dengan tema yang sama pada kasus yang berbeda telah dilakukan oleh beberapa peneliti sebelumnya. Faiz Rizky telah melakukan penelitian mengenai aplikasi pencarian tugas akhir menggunakan ontologi dan cosine similarity. Basis pengetahuan ontologi pada penelitian ini yakni mengenai Teknik Informatika. Ontologi digunakan untuk mencari hubungan antara kata yang dimasukkan oleh pengguna dan melakukan perluasan query. Cosine similarity juga diterapkan untuk memberikan bobot pada setiap dokumen yang didapatkan, sehingga hasil pencarian berupa perangkingan bobot dokumen yang relevan dengan kata kunci.

Penelitian yang dilakukan oleh Adhie dan Anita menerapkan semantic search pada OPAC menggunakan ontologi. Ontologi digunakan untuk memberikan kesamaan pemahaman makna kata antara mesin dan pengguna. Peneliti mengatakan bahwa dengan adanya semantik dapat menghasilkan pencarian yang efektif dan lebih relevan terhadap yang diinginkan oleh pengguna.

Tabel 2.1 Perbedaan dengan penelitian sebelumnya

Judul penelitian	Tahun	Metode yang digunakan	Penerapan
Penerapan semantic web dan semantic search pad digital library public online access catalog (Digilib-OPAC) untuk meningkatkan efektivitas pencarian	2012	Ontologi	Membangun sistem pencarian jurnal pada OPAC
Implementasi web semantik untuk aplikasi pencarian tugas akhir menggunakan ontologi dan cosine similarity	2015	Ontologi dan cosine similarity	Mambangun sistem pencarian tugas akhir dengan domain

			teknik informatika.
Ekspansi query berbasis semantik pada Online Public Access Catalog (OPAC)	2017	Ontologi dan cosine similarity	Menerapkan ekspansi query menggunakan semantik pada sistem OPAC Senayan Library Management System (SLiMS). Pencarian dilakukan terhadap buku tentang teknologi informasi

2.2 Ekspansi Query

Menyusun ulang query (*Query Reformulation*) yang dimasukkan pengguna adalah hal yang sering dilakukan pada *information retrieval* [4]. Langkah ini dilakukan untuk mengatasi ketidaksesuaian antara query yang dimasukkan pengguna dengan informasi yang ingin didapatkan.

Query reformulation yang sering digunakan adalah ekspansi query, yaitu dengan memanjangkan query pencarian dengan menambahkan beberapa *term* kedalamnya. Penambahan *term* berdasarkan hasil pencarian pada ontologi. Query yang dimasukkan oleh pengguna pada umumnya pendek sehingga ekspansi query dapat memperkaya atau memperluas *term* yang nantinya akan dicocokkan dengan *term* pada *indexing* [5].

2.3 Semantik

Semantik berasal dari bahasa Yunani *sema* (kata benda) yang berarti tanda atau lambang. Kata kerjanya adalah *semino* yang berarti menandai atau melambangkan. Kata semantik disepakati sebagai istilah yang digunakan untuk bidang linguistik yang mempelajari hubungan antara tanda-tanda linguistik dan hal-hal yang ditandainya dengan kata lain bidang studi dalam kata linguistik yang mempelajari makna dalam arti bahasa.

Semantik dalam kamus linguistik diartikan sebagai bagian dari struktur bahasa yang berhubungan dengan makna ungkapan dan juga dengan struktur makna. Definisi kedua, semantik adalah sistem dan penyelidikan makna dan arti

dalam suatu bahasa. Pengertian semantik dalam KBBI adalah ilmu tentang makna kata dan kalimat. Pengetahuan mengenai seluk-beluk dan pergeseran arti kata [6]. Semantik secara singkat dapat dikatakan sebagai cabang linguistik yang mengkaji mengenai makna dan hubungan antar kata.

2.4 Online Public Access Catalog (OPAC)

Online public access catalog atau disebut juga katalog online merupakan sistem katalog yang digunakan untuk menelusuri data katalog koleksi suatu perpustakaan atau unit informasi lainnya dan dapat diakses secara umum oleh pengguna. Umumnya pada sebuah sistem informasi perpustakaan dilengkapi dengan fitur *online public access catalog*.

Menurut Feather (1997, 330) menyatakan bahwa OPAC merupakan suatu pengkalan data dengan cantuman bibliografi yang biasanya menggambarkan koleksi perpustakaan tertentu. OPAC dapat mengakses langsung koleksi suatu perpustakaan secara online. Pengguna dapat melakukan penelusuran penulis buku, judul, subyek, kata kunci dan lainnya.

2.4.1 Senayan Library Management System

Senayan Library Management System (SLiMS) merupakan perangkat lunak sistem manajemen perpustakaan yang bersifat *open-source* di bawah lisensi GPL v3. Aplikasi web yang dikembangkan oleh tim dari Pusat Informasi dan Humas Departemen Pendidikan Nasional Republik Indonesia ini dibangun menggunakan bahasa pemrograman PHP dan basis data MySQL. *Online Public Access Catalog* (OPAC) merupakan salah satu fitur dalam SLiMS. Terdapat dua mode penelusuran yaitu pencarian sederhana (*simple search*) dan pencarian tingkat lanjut (*advanced search*). Mode penelusuran yang dikembangkan dalam penelitian ini adalah mode penelusuran sederhana (*simple search*).

Kelebihan SLiMS yaitu :

- a. Diperoleh dan digunakan secara gratis
- b. Memungkinkan pengembangan lebih lanjut
- c. Memiliki fitur kebutuhan perpustakaan yang lengkap

2.5 Ontologi

Munculnya istilah ontologi berasal dari kajian filsafat yang mempelajari tentang segala sesuatu yang nyata serta bagaimana mereka berhubungan. Menurut Gruber, ontologi merupakan spesifikasi baku dan jelas dari sebuah konseptualisasi [7]. Ontologi digunakan untuk merepresentasikan istilah-istilah yang ada dalam sebuah domain pengetahuan serta hubungan antar istilah-istilah tersebut [8]. Kini ontologi banyak diterapkan untuk mendukung web semantik.

Ontologi juga dapat diartikan sebagai teori yang menjelaskan makna dari suatu objek, properti dan relasi dari objek tersebut yang mungkin terjadi pada sebuah domain pengetahuan. Ontologi dalam bidang *Artificial Intelligence* (AI) memiliki dua pengertian yaitu, ontologi merupakan kosakata representasi yang sering digunakan untuk domain atau pembahasan tertentu, dan ontologi sebagai suatu *body of knowledge* untuk menjelaskan suatu domain.

2.5.1 Tipe Ontologi

Menurut (Amato, C. 2007) ada beberapa jenis ontologi, yaitu :

a. *Application and task ontology*

Application and task ontology merupakan ontologi yang khusus menyatakan *application and task* yang independen terhadap domain. Contoh ontologi tipe ini adalah PROTON yang digunakan untuk *knowledge management system* dan selanjutnya dikembangkan pula untuk *automatic entity recognition* dan *information extraction* dari teks.

b. *Upper-level ontology*

Upper-level ontology merupakan ontologi yang berupa suatu model umum untuk merepresentasikan apa yang ada di dunia, sangat serupa dengan apa yang diteliti dalam *philosophical ontology*. IEEE mengembangkan SUO (*Standart Upper Ontology*) namun sangat sulit untuk mencapai kesepakatan dalam menetapkan ontologi yang umum. SUMO (*Suggested Upper Merged Ontology*) dan Cys Upper Ontology (*Open Cys*) adalah beberapa kandidat untuk SUO. NASA juga mengembangkan upper level ontology yang disebut SWEET (*Semantic Web for Earth Environmental Technology*).

c. *Domain ontology*

Domain ontology merupakan ontologi yang merepresentasikan suatu domain tertentu saja. Penelitian yang mengembangkan ontologi dalam bidang kesehatan atau biologi, seperti *Gene Ontology*, *Cancer Ontology*, dan *Medical Ontology*. *Domain ontology* juga diterapkan dalam tugas akhir ini dengan mengembangkan ontologi mengenai Teknologi Informasi.

2.5.2 Komponen Ontologi

Secara umum ontologi memiliki beberapa komponen, antara lain [9]:

a. *Class*

Class merupakan penjelasan secara luas dari sebuah domain yang digambarkan dengan *class hierarchy*. *Top-down process*, *bottom-up process*, dan *combination process* merupakan beberapa pendekatan yang dapat digunakan dalam proses pengembangan *class hierarchy*. Proses *top-down* dimulai dengan definisi konsep yang paling umum menuju konsep yang paling khusus. Proses *bottom-up* dimulai dengan konsep yang paling spesifik kemudian terjadi pengelompokkan dari konsep tersebut yaitu konsep yang lebih umum. *Combination* merupakan kombinasi antara proses *top-down* dan *bottom-up*. *Class* biasa disebut juga sebagai *concept* atau domain.

b. Relasi

Relasi merupakan representasi tipe dari interaksi antara konsep dari sebuah domain.

c. *Properties*

Properties disebut juga *slot*, *rules*, atribut. *Properties* berhubungan dengan keterkaitan yang dimiliki suatu *class* dengan *class* yang lain. *Properties* dibedakan menjadi dua yaitu *object properties* dan *data properties*. *Object properties* adalah relasi yang dimiliki oleh suatu *class* dengan *class* yang lain, sedangkan *data properties* adalah relasi yang menunjuk kepada nilai atau tipe data yang dimiliki oleh sebuah *class*.

2.6 RDF

Resource Description Framework (RDF) merupakan salah satu bahasa yang dapat digunakan untuk pendeskripsian informasi yang terkandung dalam web, sehingga RDF dapat berperan sebagai bahasa yang digunakan untuk membentuk metadata. Elemen dalam dokumen RDF terdiri dari subjek, predikat, dan objek. Subjek mengacu pada resource yang ingin dideskripsikan. Predikat menggambarkan kelakuan atau karakteristik dari resource tersebut dan mengekspresikan hubungan antara subjek dengan objek. Objek merupakan nilai dari sebuah predikat.

RDF *properties* merupakan atribut dari sebuah sumber daya sehingga dapat merepresentasikan hubungan antara sumber daya. RDF Data Model dapat disusun dari sebuah diagram *entity-relationship*, tetapi tidak dapat menyediakan mekanisme untuk mendeskripsikan serta menjelaskan hubungan antara *properties* tersebut dengan sumber yang lainnya.

Skema RDF yang merupakan kosakata penjelasan memperbolehkan perancang kosakata untuk merepresentasikan *classes* dan *properties*. Tabel 2.2 dan tabel 2.3 merupakan sebagian dari kosakata dasar dalam RDF digunakan bersama kosakata tambahan dari model RDF dan sintaksis khusus untuk *class* dan *properties*.

Tabel 2.2 Kosakata Class RDF

Nama Class	Pengertian
rdfs:Resources	<i>Class</i> sumber daya
rdfs:Class	Konsep <i>class</i>
rdf:Property	Konsep dari <i>property</i>
rdfs:Literal	Merepresentasikan bagian dari nilai literal
rdf:Statement	Class dari statement RDF

Tabel 2.3 Kosakata Properties RDF

Nama Property	Pengertian	Domain	Range
rdfs:isDefinedBy	Nama space dari sumber	rdfs:Resource	rdfs:Resource
rdf:Subject	Subyek dari statement RDF	rdf:Statement	rdfs:Resource
rdf:Predicate	Predikat dari statement RDF	rdf:Statement	rdf:Property

2.7 SPARQL

Simple Protocol and RDF Query Language (SPARQL) merupakan bahasa query yang digunakan untuk memperoleh informasi dari graph RDF atau OWL. SPARQL menyediakan fasilitas untuk mengekstrak informasi dalam bentuk URI, *blank node* dan literal, mengesktrak subgraf RDF, dan membangun graf RDF berdasar pada informasi dari graf yang di query.

Query SPARQL didasarkan pada pencocokan pola pada graf. Pola graf yang paling sederhana adalah *triple pattern* yang mirip dengan RDF *triple*, hanya saja pola pada query dimungkinkan pemberian nama diluar terminologi RDF pada posisi subyek, predikat, dan obyek.

Klausa yang digunakan dalam query SPARQL diantaranya [10]:

1. PREFIX

Statement PREFIX merupakan sebuah meode yang digunakan sebagai penunjuk yang membawa informasi dalam suatu halaman web. PREFIX digunakan untuk menyingkat sebuah *resource*, dalam hal ini dapat diwakili oleh URI (*Uniform Resource Identifier*).

2. SELECT

Statement SELECT didefinisikan sebuah daftar variabel-variabel yang akan dikembalikan sebagai hasil dari eksekusi query dimana setiap variabel diawali dengan notasi (?).

3. WHERE

Statement WHERE didefinisikan sederetan *triple pattern* yang harus dimiliki oleh setiap hasil query yang valid. Seluruh pola yang merepresentasikan suatu kalimat RDF harus sesuai dengan RDF *triple*, yaitu terdiri dari subyek, predikat dan obyek. Ketiga RDF *triple* tersebut direpresentasikan oleh URI atau sebuah variabel dan literal.

4. OPTIONAL

Statement OPTIONAL digunakan untuk mengatasi ketidak cocokan struktur pola query dengan pola yang ada pada graf RDF.

2.8 Pembobotan TF-IDF

Metode pembobotan yang paling umum terhadap suatu *term* (*term wighting*) adalah dengan menggunakan frekuensi kemunculan *term* (kata)

atau *term frequency* (TF) yang bersangkutan pada suatu dokumen dan representasi *inverse Document Frequency* (IDF) untuk *term-term*, yang didefinisikan sebagai logaritma dari rasio jumlah keseluruhan dokumen yang diproses dengan jumlah dokumen yang memiliki *term* bersangkutan. *Term-term* yang tingkat kemunculannya jarang akan memiliki nilai IDF yang tinggi.

Rumus yang digunakan untuk menyatakan bobot (w) masing-masing dokumen terhadap kata kunci adalah:

$$W_{dt} = tf_{dt} * IDF_t \quad (1)$$

dimana :

d : dokumen ke- d

t : kata ke- t dari kata kunci

W : bobot dokumen ke- d terhadap kata ke- t

tf : banyaknya kata yang dicari pada sebuah dokumen

IDF : Inverse frekuensi dokumen $\log_{10} \left(\frac{n}{df} \right)$

n : jumlah dokumen dan query

df : banyak dokumen yang mengandung kata yang dicari

2.9 Cosine Similarity

Cosine Similarity adalah salah satu metode yang diterapkan untuk membandingkan dokumen. Cosine similarity menggunakan dua parameter vektor. Vektor tersebut didasarkan pada jumlah kemiripan kata pada dokumen [2]. Persamaan cosine similarity ditunjukkan pada rumus berikut :

$$Sim(D, D_i) = \cos(\theta) = \frac{\sum w_{q,j} w_{i,j}}{\sqrt{\sum w_{q,j}^2} \sqrt{\sum w_{i,j}^2}} \quad (2)$$

dimana :

D : dokumen acuan

D_i : dokumen ke- i

$W_{q,j}$: bobot term j pada dokumen acuan

$W_{i,j}$: bobot term j pada dokumen i

2.10 Pengantar Preprocessing Data

Proses Preprocessing dilakukan untuk mengolah data mentah menjadi data yang siap untuk diolah. Data berupa dokumen ataupun dalam bentuk paragraf akan susah diolah dalam suatu sistem. Preprocessing berperan untuk menyerap kata-kata penting yang dibutuhkan untuk proses *indexing* dalam suatu sistem. Preprocessing yang akan dilakukan pada data yaitu *case folding*, *stop word removal* dan *tokenizing*.

2.10.1 Case Folding

Case Folding merupakan suatu proses untuk mengubah semua huruf yang berada dalam suatu dokumen menjadi huruf kecil.

2.10.2 Tokenizing

Tokenizing merupakan proses penguraian deskripsi yang semula berupa kalimat menjadi per-kata baik itu berupa kata penting maupun kata-kata yang tidak penting.

2.10.3 Stop Words Removal

Stop words merupakan kata umum yang biasanya muncul dalam sebuah kalimat atau paragraf maupun dokumen, tahapan ini melakukan pengambilan kata – kata yang penting pada sebuah dokumen maupun paragraf.

2.11 Metode Pengujian

Metode pengujian ini dilakukan uji coba terhadap pengembangan yang telah dibuat. Pengujian yang dilakukan adalah mengukur nilai *precision* dan nilai *recall* terhadap hasil pencarian. Skenario pengujian dilakukan dengan memasukkan kata kunci pencarian buku.

2.11.1 Precision

Precision merupakan metode pengujian dengan melakukan perbandingan jumlah dokumen relevan terkenali yang didapatkan sistem dengan jumlah seluruh dokumen yang diambil oleh sistem yang relevan maupun kurang relevan.

Persamaan *precision* yaitu [11]:

$$Precision = \frac{Jumlah\ yang\ didapat\ relevan}{Jumlah\ yang\ didapat} \times 100\% \quad (3)$$

Pengujian ekspansi query menggunakan 3 variasi kata kunci yang berbeda sehingga perlu dihitung nilai rata-rata *precision* dari ketiga variasi tersebut. Persamaan rata-rata *precision* yaitu :

$$Rata-rata\ Precision = \frac{precision\ 1 + precision\ 2 + precision\ 3}{3} \quad (4)$$

2.11.2 Recall

Recall merupakan metode pengujian dengan melakukan perbandingan jumlah dokumen relevan terkenali yang didapatkan sistem dengan jumlah seluruh dokumen relevan yang ada pada basis data. Persamaan *recall* yaitu [11]:

$$Recall = \frac{Jumlah\ yang\ didapat\ relevan}{Jumlah\ data\ relevan\ sebenarnya\ pada\ basis\ data} \times 100\% \quad (5)$$

Pengujian ekspansi query menggunakan 3 variasi kata kunci yang berbeda sehingga perlu juga dihitung nilai rata-rata *recall* dari ketiga variasi tersebut. Persamaan rata-rata *recall* yaitu :

$$Rata-rata\ Recall = \frac{recall\ 1 + recall\ 2 + recall\ 3}{3} \quad (6)$$